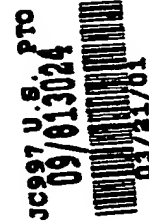


#2

Docket No. 826.1715

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
)
 Takeshi SHIMOYAMA, et al.)
) Group Art Unit: Unassigned
 Serial No.: To be assigned)
) Examiner: Unassigned
 Filed: March 20, 2001)
)
 For: COMPUTING APPARATUS USING AN)
 SPN STRUCTURE IN AN F FUNCTION)
 AND A COMPUTATION METHOD)
 THEREOF)



**SUBMISSION OF CERTIFIED COPIES OF PRIOR FOREIGN
APPLICATIONS IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith certified copies of the following foreign applications:

Japanese Patent Application No. 2000-212813
Filed: July 13, 2000; and
Japanese Patent Application No. 2000-212814
Filed: July 13, 2000.

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 20, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W., Suite 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following
application as filed with this Office.

Date of Application: July 13, 2000

Application Number: Patent Application
No. 2000-212813

Applicant(s): FUJITSU LIMITED

December 22, 2000

Commissioner,
Patent Office Kozo Oikawa

Certificate No. 2000-3105868

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

JC997 U.S. PRO
09/813024
03/21/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application: 2000年 7月13日

出 願 番 号

Application Number: 特願2000-212813

出 願 人

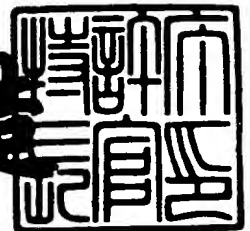
Applicant (s): 富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年12月22日

特許庁長官
Commissioner,
Patent Office

及川耕造



【書類名】 特許願

【整理番号】 0051298

【提出日】 平成12年 7月13日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明の名称】 F関数内部にS P N構造を用いた演算装置および演算方法

【請求項の数】 10

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 下山 武司

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 伊藤 孝一

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 武仲 正彦

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 鳥居 直哉

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 矢嶋 純

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 屋並 仁史

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

【氏名】 横山 和弘

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100074099

【住所又は居所】 東京都千代田区二番町 8 番地 2 0 二番町ビル 3 F

【弁理士】

【氏名又は名称】 大菅 義之

【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾 7 - 2 5 - 2 8 - 5 0 3

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 F関数内部にSPN構造を用いた演算装置および演算方法

【特許請求の範囲】

【請求項1】 複数のSボックスと線形変換部とを備えるSPN構造をF関数の内部に用いた演算装置において、

該演算装置に与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取るビット数集合入力手段と、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 A_T を出力する線形変換部存在可能性指示数値出力手段とを備えることを特徴とするF関数内部にSPN構造を用いた演算装置。

【請求項2】 前記線形変換部存在可能性指示数値出力手段が、

前記集合 T の要素から任意の k 個を選んで生成した集合の要素の和の最小値 u_k ($k = 1, 2, \dots, r$)を求める最小値決定手段と、

集合 T の要素から任意の k 個を選んで生成した集合の要素の和の最大値 v_k ($k = 1, 2, \dots, r$)を求める最大値決定手段とを更に備え、

数値 k に対して $u_k \geq v_{k'}$ ($k' = 0, 1, \dots, r, v_0 = 0$)を満たす k' の最大値を k から減算した値を w_k ($k = 1, 2, \dots, r$)とし、 w_k の最大値を $(r+1)$ の値から減算して前記 A_T の値を求めることを特徴とする請求項1記載のF関数内部にSPN構造を用いた演算装置。

【請求項3】 前記演算装置において、

前記 A_T の値が正か正でないかを判定し、正である時前記適切な線形変換部が存在すると判定する線形変換部存在判定手段を更に備えることを特徴とする請求項1、または2記載のF関数内部にSPN構造を用いた演算装置。

【請求項4】 前記演算装置において、

前記線形変換部が存在すると判定された時、該線形変換部として、前記ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成する擬似MDS行列生成手段を更に備えることを特徴とする請求項3記載のF関数内

部に S P N 構造を用いた演算装置。

【請求項 5】 前記擬似 M D S 行列生成手段が、要素が 0、または 1 の t_{ij} 行、 t_{ij} 列の部分行列 M_{ij} を要素として、 r 行、 r 列に並べた行列を $M = (M_{ij})$ ($i = 1, 2, \dots, r, j = 1, 2, \dots, r$) として、 $e = 1$ から ($A_T - 1$) までの各正数に対して $c(e) = e + r - A_T + 1$ を求め、集合 T の要素を e 個任意に選んだ集合 $T_1 = \{t_{i1}, t_{i2}, \dots, t_{ie}\}$ と、要素を $c(e)$ 個任意に選んだ集合 $T_2 = \{t_{j1}, t_{j2}, \dots, t_{jc(e)}\}$ を求め、該集合 (T_1, T_2) に対応する任意のあらゆる M の小行列、および集合 (T_2, T_1) に対応する任意のあらゆる M の小行列の階数の値が、それぞれ自小行列の行数、または列数のいずれかに等しい行列 M を求めることを特徴とする請求項 4 記載の F 関数内部に S P N 構造を用いた演算装置。

【請求項 6】 前記集合 (T_1, T_2) に対応する小行列は、前記行列 $M = (M_{ij})$ を構成する前記 r 行、 r 列の要素としての部分行列 M_{ij} のうちで、前記 $t_{i1}, t_{i2}, \dots, t_{ie}$ にそれぞれ対応する行と、 $t_{j1}, t_{j2}, \dots, t_{jc(e)}$ にそれぞれ対応する列とによって指定される部分行列によって構成されることを特徴とする請求項 5 記載の F 関数内部に S P N 構造を用いた演算装置。

【請求項 7】 複数の S ボックスと線形変換部とを備える S P N 構造を F 関数内部に用いる演算方法において、

与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取り、

該分割されたビット数をそれぞれ入・出力ビット数とする複数の S ボックスに対応する適切な線形変換部の存在可能性を示す値 A_T を出力することを特徴とする F 関数内部に S P N 構造を用いた演算方法。

【請求項 8】 前記演算方法において、

前記 A_T の値が正か正でないかを判定し、

正である時前記適切な線形変換部が存在すると判定することを特徴とする請求項 7 記載の F 関数内部に S P N 構造を用いた演算方法。

【請求項 9】 前記線形変換部が存在すると判定された時、

該線形変換部として、前記ビット数分割が均等に行われた場合の M D S 行列に対

応する擬似MD S 行列を生成することを特徴とする請求項 8 記載の F 関数内部に S P N 構造を用いた演算方法。

【請求項 1 0】 複数の S ボックスと線形変換部とを備える S P N 構造を F 関数内部に用いた演算を実行する計算機によって使用される記憶媒体において、

与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取るステップと、

該分割されたビット数をそれぞれ入・出力ビット数とする複数の S ボックスに対応する適切な線形変換部の存在可能性を示す値 A_T を出力するステップとを計算機に実行させるためのプログラムを格納した計算機読出し可能可搬型記憶媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は共通鍵ブロック暗号化方式に係り、さらに詳しくは Feistel 構造と呼ばれる構造内の F 関数の内部で用いられる複数の S ボックスに対する入出力ビット数が、複数の S ボックスの間で同一でない場合に、複数の S ボックスの後に備えられる線形変換部として、データ拡散性のよい線形変換部を生成し、その線形変換部を用いて入力データに対する暗号化を行う暗号化装置、および暗号化方法に関する。

【0 0 0 2】

【従来の技術】

高度情報化社会を迎え、情報セキュリティの確保は緊急の課題となっている。情報セキュリティの基本となるのはデータの暗号化である。高度情報化社会において高速、かつ安全な通信を実現するために、共通鍵ブロック暗号は不可欠の技術である。この共通鍵ブロック暗号のアルゴリズムとして、例えば応用分野に応じて様々な方式が提案されているが、その 1 つとして Feistel 構造と呼ばれる単純な繰返し構造のアルゴリズムがある。

【0 0 0 3】

図 1 5 はこの Feistel 構造が 1 6 段繰り返された D E S 暗号方式の説明図であ

る。同図において入力P、例えば64ビットは右側32ビットと左側32ビットとに分割され、右側32ビットはF関数51と呼ばれる非線形関数に入力され、その出力と左側32ビットとがXOR52によって排他的論理和がとられ、その結果は右側32ビットとして次の段に与えられ、次の段への左側32ビットとしては入力64ビットのうち右側32ビットがそのまま与えられる。

【0004】

図16は図15内のF関数51の構成例である。入力、例えば32ビットはビット拡張部E61によって48ビットに拡張され、XOR62によってその48ビットと鍵 K_i 48ビットとの排他的論理和がとられ、その出力は6ビットずつに分割されてSボックスと呼ばれる非線形関数にそれぞれ入力される。各Sボックス63の出力は、例えば4ビットとされ、合計32ビットが線形関数P64に入力され、データの拡散が行われる。このような構造は一般にSPN（サブステイテーションパーミュテーションネットワーク）構造と呼ばれる。

【0005】

Sボックスは暗号装置の非線形の攪拌出力を得るため、またSボックスに行われる線形関数Pは、Sボックスによる局所的な非線形出力をデータ全体に拡散させるために用いられるが、暗号装置に組み込まれる上で拡散性のよい線形変換とは何か、具体的にどう求めるか、という研究が従来より行われている。一般に、暗号に用いられる線形変換としては、一つのSボックスの出力が次のラウンドにおいて、出来る限り多くのSボックスの入力に関係することが望ましいが、現在ではより拡張された線形関数として、次のような性質を満たすものがよいとされている。すなわち、Sボックスの入出力ビット数 s に対して、線形変換Pの入力 X および出力 Y を s ビット単位、 t 個のブロック $X = (x_1, \dots, x_t)$, $Y = (y_1, \dots, y_t)$, (各 x_i, y_j は s ビット)に分割した場合、

$$Y = P(X)$$

の入出力間で成立する任意の線形関係式

$$f(x_1, \dots, x_t, y_1, \dots, y_t) = 0$$

には、入出力 x_i, y_j 合わせて $2t$ 個の変数のうち、 $t+1$ 個以上の変数が含

まれている（＝係数が0ではない）というものである。

【0006】

このような性質を満たす線形変換PとしてMDS変換が知られている。この変換は線形変換Pにおけるデータの拡散性を定義する1つの概念としての分岐数を最大とする線形変換である。この分岐数は暗号に対する差分攻撃や、線形攻撃に対する強度を評価するパラメータであり、その詳細については次の文献で説明されている。

【0007】

文献) 共通鍵ブロック暗号の選択/設計/評価に関するドキュメント、通信・放送機構

図17はそのようなMDS変換を実現する線形関数Pの説明図である。同図においては、4つのSボックス71へのそれぞれの入力、および出力は8ビットであり、合計32ビットが入力xとして線形関数Pに与えられるものとする。線形関数Pへの入力x、および出力yを、それぞれSボックスに対応させて、8ビットずつに分割した変数 x_i ($i = 1 \sim 4$)、 y_j ($j = 1 \sim 4$)とする。

【0008】

ここで x_i に入力差分 Δx_i が与えられた時、そのiの集合を次のように書き、これを入力アクティブSボックスと名付ける。

$$\{i \mid \Delta x_i \neq 0\}$$

例えば x_1 、 x_2 に入力差分が与えられた時、この集合は $\{1, 2\}$ となる。

【0009】

この入力アクティブSボックスに対応して、出力差分 Δy_j が生じる y_j に対応して、次の集合を出力アクティブSボックスと名付ける。

$$\{j \mid \Delta y_j \neq 0\}$$

これら2つの集合の和集合

$$\{i \mid \Delta x_i \neq 0\} \cup \{j \mid \Delta y_j \neq 0\}$$

をアクティブSボックスと名付ける。

【0010】

そしてこの集合アクティブSボックスの要素の数 $act S(P)$ の最小値は、

線形変換 P によって決定される。アクティブ S ボックスの要素の数の最小値

$$\min (act S (P))$$

をアクティブ S ボックスの数と呼ぶことにする。このアクティブ S ボックスの数の最大値は、前述の線形関係式に含まれる変数の数 $(t + 1)$ に一致するとされている。このアクティブ S ボックスの数の最大値が、例えば 5 となる線形変換 P が存在するとすれば、入力 x_i ($i = 1 \sim 4$) の 1 個が変化すると、出力 y_j ($j = 1 \sim 4$) の 4 個が変化することになり、また出力の 1 個は入力の 5 個から影響されることになる。

【0 0 1 1】

図 1 8 はこのような MDS 変換に相当する MDS 行列の説明図である。同図において MDS 行列は、それぞれ例えば 0、または 1 の要素からなる 8 行、8 列の部分的な行列 a_{ij} ($i = 1 \sim 4, j = 1 \sim 4$) から構成されている。この a_{ij} の行数と列数は、図 1 7 で説明した S ボックス 7 1 の入出力ビット数に対応する。

【0 0 1 2】

このような MDS 行列が持つべき性質を説明する。図 1 8 の行列が、 MDS 行列として図 1 7 で説明した線形関数 P に望まれる高い拡散性を有するためには、部分的な行列 a_{ij} を要素として考えた場合の 4 行、4 列の全体から、行数と列数が等しい任意の小行列を選択した時に、その全ての小行列が正則であることが必要とされている。

【0 0 1 3】

すなわち例えば 1 行と 1 列を指定した $(1, 1)$ 小行列、2 行と 2 列を指定した $(2, 2)$ 小行列、3 行と 3 列を指定した $(3, 3)$ 小行列、および行列全体と一致する $(4, 4)$ 小行列の全てが、逆行列を持ち、同じ配置の行列式が 0 でなく、ランク (階数) がフルであるという性質を持つものとされている。

【0 0 1 4】

【発明が解決しようとする課題】

このように共通鍵ブロック暗号化方式における Feistel 構造内の F 関数の中で、データの拡散に重要な役割を果たす線形変換 P としての MDS 行列の設計は、従来は複数の S ボックスの入出力サイズが等しいことを前提として行われていた

が、複数の S ボックスの間で入出力サイズが異なる場合については、適切な線形変換 P が存在するか否か、存在するとすればその変換をどのように構成すればよいかについては、従来全く知られていないという問題点があった。

【 0 0 1 5 】

本発明の課題は、上述の問題点に鑑み、複数の S ボックスの間で入出力サイズが異なる場合に、データの拡散性に優れた線形変換が存在するか否かを判定し、そのような線形変換が存在する場合に、そのような線形変換に相当する擬似 MD S 行列を生成し、それを用いて入力データに対応する暗号文を生成する暗号文生成装置、および生成方法を提供することである。

【 0 0 1 6 】

【課題を解決するための手段】

図 1 は本発明の演算装置の原理構成ブロック図である。同図は、Feistel 構造の F 関数の内部に、複数の S ボックスと線形変換部とを備える演算装置 1 の原理構成ブロック図である。

【 0 0 1 7 】

図 1 においてビット数集合入力手段 2 は、演算装置 1 に与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取るものである。

【 0 0 1 8 】

また線形変換部存在可能性指示数値出力手段 3 は、分割されたビット数をそれぞれ入・出力ビット数とする複数の S ボックスに対応して、データ拡散性に優れた線形変換部の存在可能性を示す値、例えばアクティブ S ボックスの数の最大値 A_T を出力するものである。

【 0 0 1 9 】

本発明の実施の形態においては、この A_T の値が正である時、適切な線形変換部が存在すると判定する線形変換部存在判定手段 4 を更に備えることも、またそのような線形変換部として、ビット数分割が均等に行われた場合の MD S 行列に対応する擬似 MD S 行列を生成する擬似 MD S 行列生成手段 5 を更に備えることもできる。

【 0 0 2 0 】

また発明の実施の形態においては、線形変換部存在可能性指示数値出力手段 3 が、前述のビット数集合の要素から任意の k 個を選んで生成した集合の要素の和の最小値 u_k ($k = 1, 2, \dots, r$) を求める最小値決定手段と、同様に k 個を選んで生成した集合の要素の和の最大値 v_k を求める最大値決定手段とを更に備え、数値 k に対して $u_k \geq v_{k'}$ ($k' = 0, 1, \dots, r, v_0 = 0$) を満たす k' の最大値を k から減算した値を w_k とし、 w_k の最大値を $(r + 1)$ の値から減算して A_T の値を求めることもできる。

【 0 0 2 1 】

更に本発明の実施の形態においては、擬似 MD S 行列生成手段 5 は、要素が 0、または 1 の t_i 行、 t_j 列の部分行列 M_{ij} を要素として r 行、 r 列に並べた行列を $M = (M_{ij})$ ($i, j = 1, 2, \dots, r$) とし、 $e - 1$ から $(A_T - 1)$ までの各正数に対して、 $c(e) = e + r - A_T + 1$ を求め、集合 T の要素を e 個任意に選んだ集合 T_1 と、要素を $c(e)$ 個任意に選んだ集合 T_2 を求め、その集合 (T_1, T_2) に対応する任意のあらゆる M の小行列、および集合 (T_2, T_1) に対応する任意のあらゆる M の小行列の階数の値がそれぞれ小行列の行数、または列数のいずれかに等しい行列 M を求めることもできる。

【 0 0 2 2 】

この時、例えば集合 (T_1, T_2) に対応する小行列は、前述の部分行列 M_{ij} のうちで、集合 T_1 の各要素にそれぞれ対応する行と、集合 T_2 の各要素に対応する列によって指定される部分行列によって構成されることもできる。

【 0 0 2 3 】

本発明の演算方法として、複数の S ボックスと線形変換部とを備える SPN 構造を F 関数の内部に用いる演算方法において、与えられる入力データのビット数を非均等に分割したビット数の集合 T の入力を受け取り、分割されたビット数をそれぞれ入・出力ビット数とする複数の S ボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブ S ボックスの数の最大値 A_T を出力する方法が用いられる。

【 0 0 2 4 】

この方法においては、発明の実施形態では、 A_T の値が正である時、適切な線形変換部が存在すると判定することもでき、またそのような線形変換部として、ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成することもできる。

【0025】

更に本発明においては、複数のSボックスと線形変換部とを備えるSPN構造を、F関数の内部に用いた演算を実行する計算機によって使用される記憶媒体として、与えられる入力データの全ビット数を非均等に分割したビット数集合Tの入力を受け取るステップと、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値 A_T を出力するステップとを計算機に実行させるためのプログラムを格納した計算機読出し可能可搬型記憶媒体が用いられる。

【0026】

以上説明したように、本発明によればFeistel 構造の内部のF関数を構成するSPN構造内で、複数のSボックスの入出力ビット数が非均等の場合に対して、データの拡散性に優れた線形変換部の生成が可能となる。

【0027】

【発明の実施の形態】

本発明においては、Feistel 構造の内部に備えられるF関数を構成するSPN構造内で、複数のSボックスの間で入出力ビット数が全て同じではない場合の暗号化アルゴリズム、およびそのアルゴリズムを用いた暗号化装置を本発明の実施形態として説明する。

【0028】

図2はそのような暗号化装置の構成ブロック図である。同図において暗号化装置は処理装置10、入力ファイル11、出力ファイル12、表示装置13、および入出力装置14によって構成されている。

【0029】

入力ファイル11には、例えば暗号化の対象としての平文、Feistel 構造内の

F関数への入力データのビット数 n 、ビット数 n が複数のSボックスに入力される場合の各Sボックスへの入力ビット数 t_1, t_2, \dots, t_r を要素とする集合 T などが格納されている。

【0030】

処理装置10の内部には、入力ファイル11に格納されている集合 T の内容を用いて、複数のSボックスに対するそれぞれの入出力ビット数が同じでない場合に、その複数のSボックスの出力に対応する適切な線形変換部の存在可能性を示す数値 A_T を計算する A_T 計算部15、計算された A_T の値を用いてそのような線形変換部が存在するか否かを判定する線形変換部存在判定部16、そのような線形変換部が存在すると判定された時に、そのような変換部としての擬似MDS行列を計算する擬似MDS行列生成部17、生成された擬似MDS行列を用いて、入力ファイル11に格納されている平文に対する暗号文を生成する暗号文生成部18などを備えている。

【0031】

出力ファイル12には、 A_T 計算部15によって計算された A_T の値、擬似MDS行列、およびその擬似MDS行列を用いた暗号化アルゴリズムなどが格納される。

【0032】

図3は本実施形態において用いられるF関数の内部のSPN構造の例である。入力データ32ビットは、例えば6, 5, 5, 5, 5、および6ビットに分割され、非線形変換部としてのそれぞれのSボックス21に入力される。各Sボックスは入力ビット数と同じ出力ビット数を持ち、各Sボックスの出力は合成されて32ビットとして線形変換部P22に与えられ、その変換結果がF関数の出力となる。

【0033】

本実施形態においては、このように複数のSボックスへの入出力ビット数が同じでない場合に、そのビット分割の仕方によって適切な線形変換部Pが存在するか否か、また存在する場合にはその線形変換部をどのように求めるかが発明のポイントとなる。

【 0 0 3 4 】

ここで入力データのビット数 n を非均等に分割する理由について説明する。例えば従来技術で説明した図 1 7 では、入力 3 2 ビットを 4 つに分割した 8 ビットずつが、4 つの S ボックス 7 1 に入力されている。このような S ボックスは一般的には演算の高速化のために計算機の一次キャッシュメモリにテーブルとして格納され、そのテーブルにアクセスすることによって演算が行われる。図 1 7 ではテーブルは 4 つであり、4 回のテーブルアクセスが必要となる。

【 0 0 3 5 】

これに対して本実施形態では、図 3 に示すように例えば、入力 3 2 ビットが 6, 5, 5, 5, 5、および 6 ビットの 6 つに分割され、6 個の S ボックスにそれぞれ入力される。このようにビット数の少ない 6 個の S ボックスに分割すると、それぞれの S ボックスに対応するテーブルのサイズは小さくなり、一次キャッシュメモリ量の少ない計算機を使用しても、演算を実行することが可能となる。

【 0 0 3 6 】

最近の計算機の一次キャッシュメモリ量の増大の傾向につれて、1 つのテーブルサイズを大きくしてテーブルアクセスの回数を減らし、演算を高速化する可能性が開かれている。そこで本実施形態においては、計算機の一次キャッシュメモリ量に対応して、ビット数分割を変更できるようなビット数分割法を用いることにする。

【 0 0 3 7 】

前述のように 3 2 ビットを 8 ビット \times 4 に分割している場合には、テーブルの数を 3 つにするためには 8, 1 6, 8 ビットというような分割に変換するしか方法がなく、1 6 ビット入力の S ボックスに対しては 2^{16} 個の領域を持つテーブルが必要となってしまうことになる。それに対して図 3 の分割方法では、例えば 2 個ずつまとめて 1 1, 1 0, 1 1 ビットの 3 つに分割することもでき、 2^{11} 個の領域を持つテーブルを計算機の一次キャッシュメモリに格納することができれば、演算の高速化が可能となる。

【 0 0 3 8 】

図 4 は本実施形態における暗号文生成処理の全体フローチャートである。同図

において処理が開始されると、まずステップ S 1 で図 2 で説明した線形変換部が存在するか否かを判定するための数値 A_T が求められる。この数値 A_T としては、前述のアクティブ S ボックスの要素の数の最小値の最大値が用いられる。以後この A_T を“アクティブ S ボックス数の最大値”と呼ぶ。

【0039】

そしてステップ S 2 で、求められた A_T の値に応じて、適切な線形変換 P が存在するか否かが判定される。具体的には A_T の値が正の数である時にはそのような線形変換が存在すると判定され、0、または負の数である時にはそのような線形変換は存在しないと判定される。

【0040】

線形変換が存在すると判定されると、ステップ S 3 でその線形変換を実現する行列、すなわち擬似 M D S 行列が生成され、ステップ S 4 でその擬似 M D S 行列を用いた暗号化アルゴリズム、すなわち Feistel 構造が生成され、ステップ S 5 でその暗号化アルゴリズムを用いて平文が暗号化されて、処理を終了する。

【0041】

ステップ S 2 で A_T の値が 0、または負の数となり、適切な線形変換が存在しないと判定されると、ステップ S 6 でエラーが発生したことを示すメッセージが出力されて、処理を終了する。

【0042】

図 5 は図 4 のステップ S 1、すなわちアクティブ S ボックス数の最大値 A_T の計算処理の詳細フローチャートである。同図において処理が開始されると、まずステップ S 1 0 で集合 T の内容が入力され、ステップ S 1 1 で集合 T を構成する r 個の要素のうちから k 個を選んで生成された集合の要素の和の最小値 u_k が $k = 0, 1, 2, \dots, r$ に対して求められる。

【0043】

続いてステップ S 1 2 で、同様に集合 T の要素から任意の k 個を選んで生成された集合の要素の和の最大値 v_k が求められる。

続いてステップ S 1 3 で k ($= 1, 2, \dots, r$) と k' ($= 0, 1, 2, \dots, r$) に対して次の不等式

$$u_k \geq v_{k'} \quad (\text{ただし } v_0 = 0)$$

を満たす k' の最大値を k から減算した値が w_k ($k = 1, 2, \dots, r$) として求められる。

【0044】

最後にステップ S 1 4 で w_k の最大値が $r + 1$ から減算され、 A_T の値として求められ、処理を終了する。

図 6 は図 4 のステップ S 3 の処理、すなわち擬似 MD S 行列生成処理の詳細フローチャートである。同図において処理が開始されると、まずステップ S 2 0 で分割ビット数の集合 T の内容に応じて、 t_i 行、 t_j 列であり、要素が 0、または 1 となっている行列 M_{ij} ($i, j = 1 \sim r$) を作り、そのような $r \times r$ 個の行列 M_{ij} を要素として r 行、 r 列に並べた行列 M がランダムに新しく選択される。図 3 で説明した F 関数の例ではこの行列 M は全体としては 3 2 行、3 2 列の行列となる。ここで M_{ij} を行列 M の部分行列と呼ぶことにする。

【0045】

続いてステップ S 2 1 で e の値が 1 に初期化され、ステップ S 2 2 で e の値がアクティブ S ボックス数の最大値 A_T から 1 を引いた値を越えたか否かが判定され、越えていない場合にはステップ S 2 3 で次式によって $c(e)$ の値が求められる。

【0046】

$$c(e) = e + r - A_T + 1$$

ステップ S 2 4 で集合 T から e 個の要素を任意に選び、集合 T_1 が新しく求められ、ステップ S 2 5 で新しい集合 T_1 が選択できたか否かが判定され、選択できた場合にはステップ S 2 6 で同様に集合 T から $c(e)$ 個の要素が任意に新しく選ばれ、集合 T_2 が求められ、ステップ S 2 7 でそのような新しい集合 T_2 が選択できたか否かが判定される。なおステップ S 2 4、および S 2 6 で新しく選択された集合 T_1 、および T_2 を以下のように記述するものとする。

【0047】

$$T_1 = \{t_{i1}, t_{i2}, \dots, t_{ie}\}$$

$$T_2 = \{t_{j1}, t_{j2}, \dots, t_{jc(e)}\}$$

ステップ S 2 7 で新しく集合 T_2 が選択できたと判定されると、ステップ S 2 8 で行列 M の小行列のうちで集合 T_1 , T_2 に対応する小行列のランク（階数）が求められる。この T_1 , T_2 に対応する小行列の意味については後述する。そしてステップ S 2 9 で求められたランクの値が 外 1 または 外 2 のどちら

【 0 0 4 8 】

【 外 1 】

$$\sum_{p=1}^e t_{ip}$$

【 0 0 4 9 】

【 外 2 】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【 0 0 5 0 】

か、すなわち行数と列数のいずれかに等しいか否かが判定される。

等しい場合には、ステップ S 3 0 で行列 M の小行列のうちで集合 T_2 , T_1 に対応する小行列のランクが求められ、ステップ S 3 1 でそのランクの値が 外 3

【 0 0 5 1 】

【 外 3 】

$$\sum_{p=1}^e t_{ip}$$

【 0 0 5 2 】

または 外 4 のいずれかに等しいか否かが判定される。

【 0 0 5 3 】

【外 4】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【0 0 5 4】

ステップ S 3 1 でランクの値が 2 つの総和（行数、列数）のいずれかに等しいと判定されると、ステップ S 2 6 に戻り $c(e)$ 個の要素が新たに選択され、新しい集合 T_2 が求められ、ステップ S 2 7 の判定以降の処理が繰り返される。

【0 0 5 5】

そしてステップ S 2 7 で $c(e)$ 個の集合 T_2 が新しく選択できなかったと判定されると、以前にステップ S 2 4 で選択された集合、すなわち e 個の要素からなる集合 T_1 に対応する処理が終了したことになるため、ステップ S 2 4 で e 個の要素からなる集合 T_1 として新しい集合が求められ、ステップ S 2 5 以降の処理が繰り返される。

【0 0 5 6】

ステップ S 2 5 で新しい集合 T_1 が選択できなかったと判定されると、ステップ S 2 1 で初期化された $e = 1$ の値に対応する処理が終了したことになるので、ステップ S 3 2 で e の値がインクリメントされ、ステップ S 2 2 以降の処理が繰り返される。

【0 0 5 7】

このような処理の間に、ステップ S 2 9 でランクの値が 2 つの総和の値のいずれにも等しくないと判定された時、およびステップ S 3 1 で同様にランクの値が 2 つの総和のいずれにも等しくないと判定された時には、ステップ S 2 0 でランダムに選択された行列 M が擬似 M D S 行列としては不適当なものであるものとして、ステップ S 2 0 で新しい行列 M をランダムに選択する処理以降の処理が繰り返され、ステップ S 2 2 で e の値が $A_T - 1$ の値を越えたと判定されると、行列 M の内容が擬似 M D S 行列として出力され、処理を終了する。

【0 0 5 8】

図 5、および図 6 で説明した処理について、具体例を用いて更に説明する。図 3 で説明した全部で 3 2 ビットの入力ビットに対する 6 個の S ボックスに対応して、分割される入出力ビット数の集合は次式で与えられる。

【 0 0 5 9 】

$$T = \{6, 5, 5, 5, 5, 6\}$$

このような集合 T に対応して前述の最小値 u_k 、および最大値 v_k ($v_{k'}$) は次のようになる。

【 0 0 6 0 】

$$(u_1, u_2, u_3, u_4, u_5, u_6) = (5, 10, 15, 20, 26, 32)$$

$$(v_0, v_1, v_2, v_3, v_4, v_5, v_6) = (0, 6, 12, 17, 22, 27, 32)$$

その結果 w_k は次式となり、その最大値は 1 となる。

【 0 0 6 1 】

$$(w_1, w_2, w_3, w_4, w_5, w_6) = (1, 1, 1, 1, 1, 0)$$

最終的にアクティブ S ボックス数の最大値 A_T は、この w_k の最大値を用いて次式によって求められる。

【 0 0 6 2 】

$$A_T = (6 + 1) - 1 = 6$$

この A_T の値が 6、すなわち正の数であることから、このように入出力ビット数が分割された 6 個の S ボックスによる非線形変換に適切な線形変換が存在するということが判定される。前述のようにこの行列 M は全体として 3 2 行、3 2 列であり、その要素が 0、または 1 のうちからランダムに選択され、選択された行列が図 6 のフローチャートによって擬似 M D S 行列の性質を満たすか否かが判定される。

【 0 0 6 3 】

従ってそのような行列 M を生成するためには、原理的には 3 2 行、3 2 列の行列の全ての要素を 0、または 1 とした場合について図 6 のフローチャートの処理を繰り返して、擬似 M D S 行列を求めればよいことになるが、その計算量は膨大

となる。

【0064】

本実施形態では計算量を削減するための擬似MDS行列生成法を用いることにするが、その方法については後述することとし、その方法によって求められた行列Mの例を図7に示す。この例の行列が図6のフローチャートの処理において、最終的にステップS33で出力されるまでの過程の最初の部分について具体的に説明する。なお図7において、行列内の実線で区切られた部分は、図6のステップS20で説明した行列M内の部分行列 M_{ij} に相当する。

【0065】

図6に対応する処理の具体例を説明する前に、まず例えばステップS28で説明した T_1 、 T_2 に対応する小行列の意味について、図8を用いて説明する。図8において、例えば集合 $T_1 = \{t_2, t_3, t_6\}$ 、 $T_2 = \{t_2, t_3, t_5, t_6\}$ とした場合には、 T_1 、 T_2 に対応する小行列として図8(a)に示される行列が生成され、そのランクが求められる。すなわちそれぞれが行列である M_{ij} を部分行列とする行列Mから計3行と4列とが指定されて小行列が構成される。この小行列はビット単位、すなわち0、または1の要素単位では16行、21列の行列となる。

【0066】

また図6のステップS30で説明した T_2 、 T_1 に対応する小行列としては行として集合 T_2 の要素である t_2 、 t_3 、 t_5 、および t_6 に相当する行と、集合 T_1 の要素としての t_2 、 t_3 、および t_6 に対応する列が選択されて小行列が構成される。この小行列を図8(b)に示す。この行列は21行、16列の行列である。

【0067】

ここで本実施形態におけるMDS変換としての擬似MDS行列が持つべき性質について説明する。 $n = 32$ ビットを6個に非均等に分割した集合の例としての前述のTに対して、アクティブSボックスの数の最大値は $A_T = 6$ となる。これに対してビット数の分割を均等に行う場合に A_T に相当する値は7であり、その差は1となる。

【0068】

前述のようにビット分割が均等な場合のMDS変換としてのMDS行列では、図8で説明したような M_{ij} （全ての行数、および全ての列数は等しい）を要素とする行列から、任意の1行と1列を指定した（1，1）小行列、2行と2列を指定した（2，2）小行列、3行と3列を指定した（3，3）小行列、・・・を考え、そのような任意の小行列が全て正則であることがMDS行列の性質として成立する。

【0069】

これに対して擬似MDS行列では、前述の差が1であることから、ビット分割が均等な場合に選択される小行列の行、または列のいずれかに1を加えた行列が小行列として選択され、任意の小行列のランクがフル、すなわち小行列のランクがその行数、または列数に等しくなるという性質がある。

【0070】

すなわち任意の（1，2），（2，1），（2，3），（3，2），（3，4），（4，3），（4，5），（5，4），（5，6）、および（6，5）の10種類の小行列のランクが、それぞれの小行列の行数、または列数に等しい行列を、擬似MDS行列として図6のフローチャートにおいて選択すべきことになる。これが本実施形態における擬似MDS行列が持つべき性質であるが、その詳細な数学的説明（証明など）については省略する。

【0071】

ここで前述の例に戻り、図6のフローチャートに対応して、そのような性質を持つ行列Mの選択の過程の最初の部分の説明を続ける。まず図6のステップS21でeの値が1とされ、ステップS23でc(e)の値として2が求められる。そしてステップS24で集合 T_1 として1個だけの要素を持つ $\{t_1\} = \{6\}$ が選択されたとする。またステップS26でc(e)、すなわち2個の要素を持つ集合 T_2 として $\{t_1, t_2\} = \{6, 5\}$ が選択されたものとする。

【0072】

図9はこの場合にそのランクが計算されるべき、ステップS28における T_1 、 T_2 に対応する行列である。すなわち図8において行としては1行目、列とし

ては 1 列目と 2 列目とが指定されることになり、小行列は M_{11} と M_{12} を成分とする行列であり、その実際の内容は図 7 から図 9 のようになる。この小行列のランクは 6 である。

【 0 0 7 3 】

このランクの値、すなわち 6 はステップ S 2 9 で 外 5 または 外 6 のい

【 0 0 7 4 】

【外 5】

$$\sum_{p=1}^e t_{ip}$$

【 0 0 7 5 】

【外 6】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【 0 0 7 6 】

いずれかの値と等しいか否かが判定される。これらの 2 つの値は図 9 の小行列の行数と列数を示し、この場合は行数、すなわち 外 7 がランクの値と等しくなり

【 0 0 7 7 】

【外 7】

$$\sum_{p=1}^e t_{ip}$$

【 0 0 7 8 】

、この小行列はフルランクであることが判定される。

図 1 0 はステップ S 3 0 でそのランクが計算されるべき T_2 , T_1 に対応する

小行列の例である。前述と同様に、ここでは図8の M_{ij} のうち、行としては1行目と2行目、例としては1列目が指定されることにより、 M_{11} と M_{21} とによって図10に示す小行列が構成される。そのランクは6であり、ステップS31でステップS29におけると同様に2つの総和と比較され、外8の値と等しいこ

【0079】

【外8】

$$\sum_{p=1}^e t_{ip}$$

【0080】

とが判定されて、以後の処理が続けられる。

そして図6のフローチャートに従って、前述の10種類の小行列の任意のものについて、各小行列のランクがフルであることが図7の32行、32列の行列に対して確認され、最終的にステップS33でこの行列Mが擬似MDS行列として出力されることになる。

【0081】

次に図7に示した擬似MDS行列の生成法について説明する。この行列を生成するためには、原理的には前述のように32行×32列の行列の全ての要素を0、または1にランダムに変化させて、図6のフローチャートを満足する行列Mを探すことになるが、その計算量は膨大となる。

【0082】

そこでより能率的な方法として、本実施形態においてはまず全ビット数を30ビットとし、30ビットを6個に分割した集合 $T = \{5, 5, 5, 5, 5, 5\}$ に対するMDS行列を従来技術によって求め、求められた30行、30列の行列に対して図7に示すように最も上の行の M_{1j} ($j = 1 \sim 6$)、最も下の行の M_{6j} ($j = 1 \sim 6$)、最も左の列の M_{i1} ($i = 1 \sim 6$)、および最も右の列の M_{i6} ($i = 1 \sim 6$)に対応してそれぞれ1行、1列の要素を追加することで、擬似MDS行列を作成することにする。

【 0 0 8 3 】

図 1 1、図 1 2 はそのような 3 0 行、3 0 列の M S D 行列を構成するための 5 行、5 列の部分行列 3 2 個を示している。この 3 2 個の部分行列はそれぞれ 5 行、5 列の行列であり、各行列には 0 ～ 3 1 の番号が付けられている。0 番の行列は図 1 1 の左上の行列であり、5 行、5 列の行列の要素は全て 0 である。5 行、5 列の下の“0”はこの行列に対応する（同じ配置の）行列式の値を示している。0 番目の行列に対しては、当然対応する行列式の値は 0 である。

【 0 0 8 4 】

例えばその下の番号 1 の行列に対する行列式の値は 1 であり、以降図 1 2 の右下の 3 1 番までの全ての行列に対する行列式の値も 1 となっている。

従来技術の方法を用いることによって図 1 1、図 1 2 に示した番号の 5 行、5 列の部分行列を 6 行、6 列に並べることによって、3 0 ビットを 6 個に均等分割した場合に対応する M D S 行列の例として図 1 3 の行列が得られる。行列内の数字は図 1 1、図 1 2 で説明した各行列の番号を表わす。

【 0 0 8 5 】

図 1 3 に示した行列は 3 0 行、3 0 列の行列であり、最も上、下の部分行列に対して 1 行、最も左、右の部分行列に対して 1 列の要素をランダムに追加し、その行列に対して図 6 で示したフローチャートの処理を実行することによって、図 7 に示した擬似 M D S 行列を比較的容易に生成することができる。

【 0 0 8 6 】

図 1 4 は本発明を実現するためのプログラムのコンピュータへのローディングの説明図である。本発明の実施形態としての暗号化装置、例えば図 2 に示したシステムなどは、当然一般的なコンピュータシステムとして構成することができる。

【 0 0 8 7 】

図 1 4 はそのようなシステムの構成を示し、コンピュータ 3 1 は本体 3 2 と、メモリ 3 3 とによって構成されている。メモリ 3 3 はランダムアクセスメモリ（RAM）、ハードディスク、磁気ディスクなどの記憶装置であり、本発明の特許請求の範囲第 1 0 項のプログラムや、図 4 ～図 6 で説明したプログラムなどはメ

モリ 3 3 に格納され、そのプログラムが本体 3 2 によって実行されることにより、本発明の擬似 M S D 行列が求められ、入力データに対する暗号化が行われる。

【 0 0 8 8 】

本発明を実現するためのプログラムは、プログラム提供者側からネットワーク 3 4 を介してコンピュータ 3 1 にロードされることも、また市販され、流通している可搬型記憶媒体 3 5 に格納され、そのプログラムがコンピュータ 3 1 にロードされることによって実現されることも可能である。可搬型記憶媒体 3 5 としてはフロッピーディスク、C D - R O M、光ディスク、光磁気ディスクなど、様々な形式の記憶媒体を使用することができる。前述のプログラムなどは、このような記憶媒体に格納され、コンピュータ 3 1 にロードされることによって、本実施形態における擬似 M S D 行列が生成され、その行列を用いて入力データに対する暗号文を生成することが可能となる。

【 0 0 8 9 】

【発明の効果】

以上詳細に説明したように、本発明によれば、F 関数の内部の複数の S ボックスの入出力サイズが同一でない場合において、適切な線形変換としての擬似 S M D 行列の存在の有無を判定することができ、そのような行列が存在する場合にはその擬似 M S D 行列を生成し、その行列を使用した暗号化を行うことによって、拡散性能に優れた暗号を生成することができ、暗号化装置の性能向上に寄与するところが多い。

【図面の簡単な説明】

【図 1】

本発明の原理構成ブロック図である。

【図 2】

本発明の実施形態としての暗号化装置のシステム構成を示すブロック図である。

【図 3】

本実施形態における F 関数の構成例を示す図である。

【図 4】

暗号文生成の全体処理フローチャートである。

【図 5】

アクティブ S ボックスの数の最大値 A_T を求める処理の詳細フローチャートである。

【図 6】

擬似 MD S 行列を求める処理の詳細フローチャートである。

【図 7】

求められた擬似 MD S 行列の例を示す図である。

【図 8】

2 つの集合に対応する小行列を説明する図である。

【図 9】

擬似 MD S 行列の小行列の例（その 1）を示す図である。

【図 10】

擬似 MD S 行列の小行列の例（その 2）を示す図である。

【図 11】

30 行 × 30 列の MD S 行列を求めるための部分行列を示す図（その 1）である。

【図 12】

30 行 × 30 列の MD S 行列を求めるための部分行列を示す図（その 2）である。

【図 13】

図 11, 図 12 の部分行列を用いた MD S 行列の例を示す図である。

【図 14】

本発明におけるプログラムのコンピュータへのローディングを説明する図である。

【図 15】

DES 暗号の基本構造を示す図である。

【図 16】

図 15 における F 関数の構成例の説明図である。

【図 1 7】

F 関数内の線形変換 P としての M D S 変換の説明図である。

【図 1 8】

M D S 変換としての M D S 行列の説明図である。

【符号の説明】

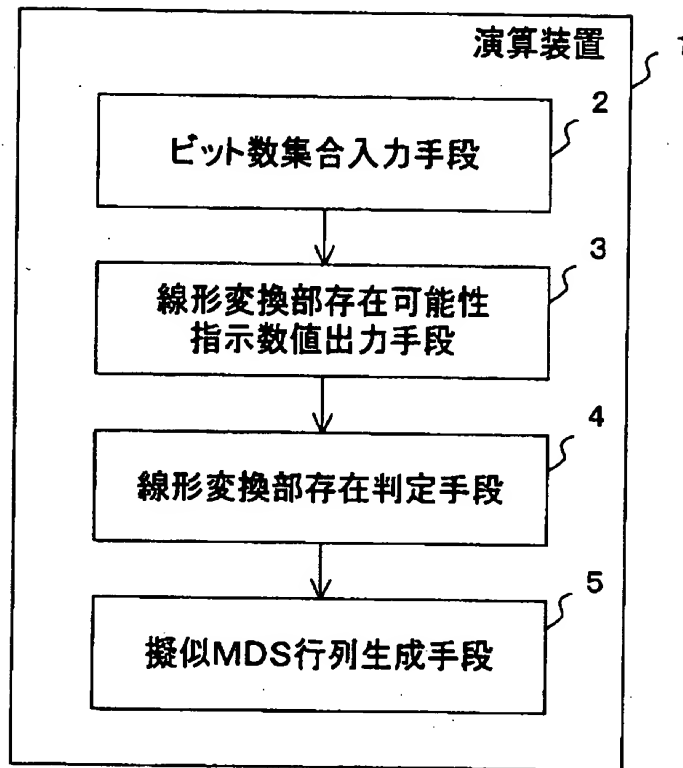
- 1 演算装置
- 2 ビット数集合入力手段
- 3 線形変換部存在可能性指示数値出力手段
- 4 線形変換部存在判定手段
- 5 擬似 M D S 行列生成手段
- 1 0 処理装置
- 1 1 入力ファイル
- 1 2 出力ファイル
- 1 3 表示装置
- 1 4 入出力装置
- 1 5 A_T 計算部
- 1 6 線形変換部存在判定部
- 1 7 擬似 M D S 行列生成部
- 1 8 暗号文生成部

【書類名】

図面

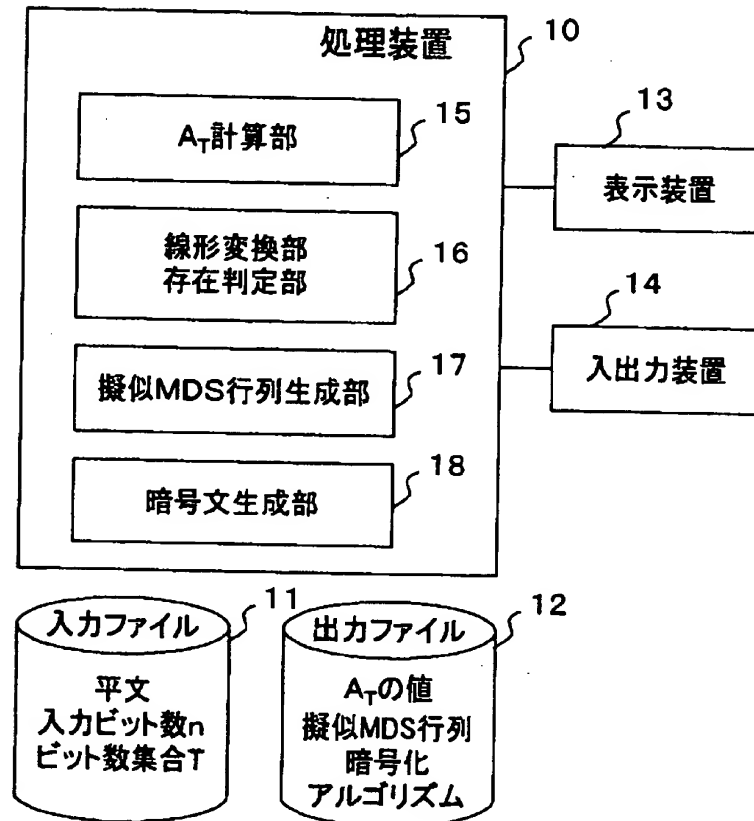
【図1】

本発明の原理構成ブロック図



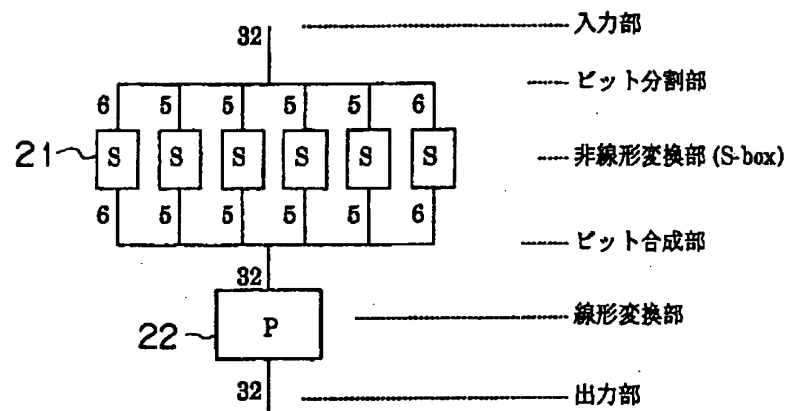
【図2】

本発明の実施形態としての
暗号化装置のシステム構成を示すブロック図



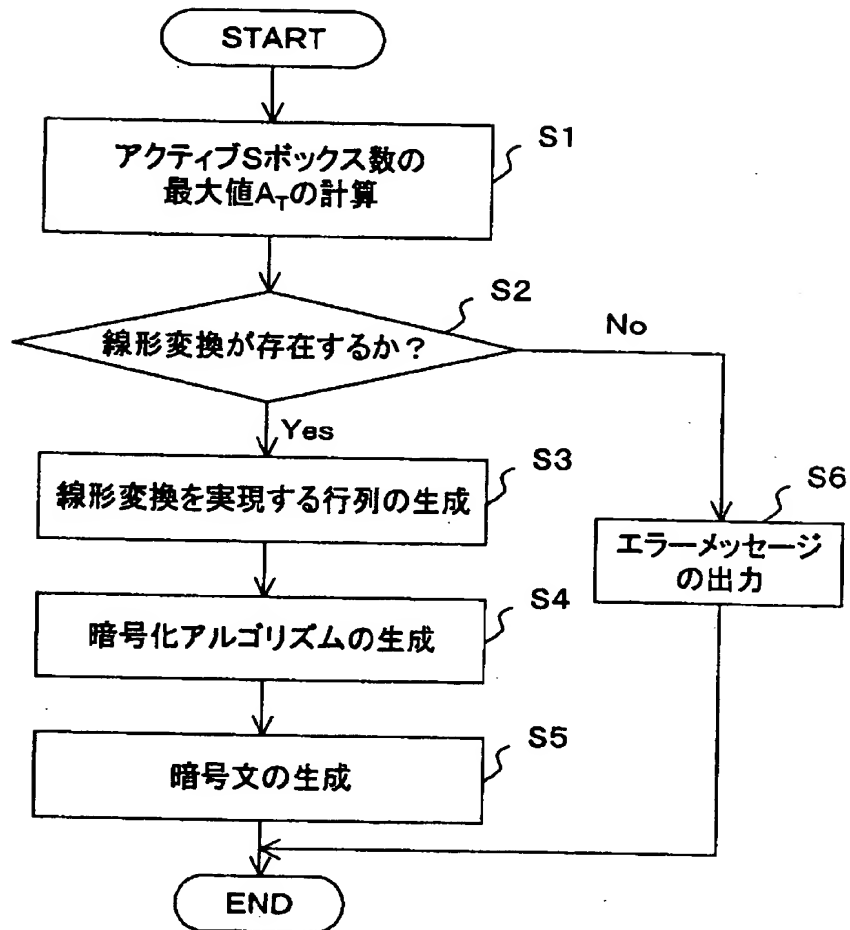
【図 3】

本実施形態における F 関数の構成例を示す図



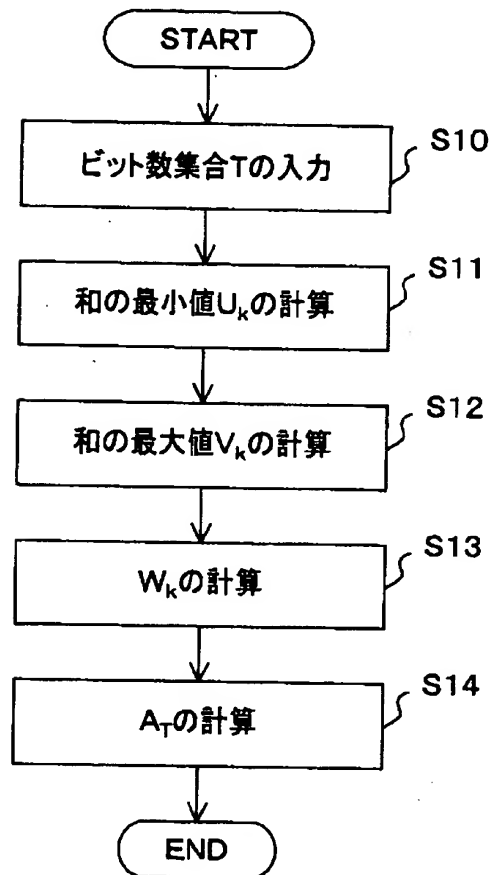
【図4】

暗号文生成の全体処理フローチャート



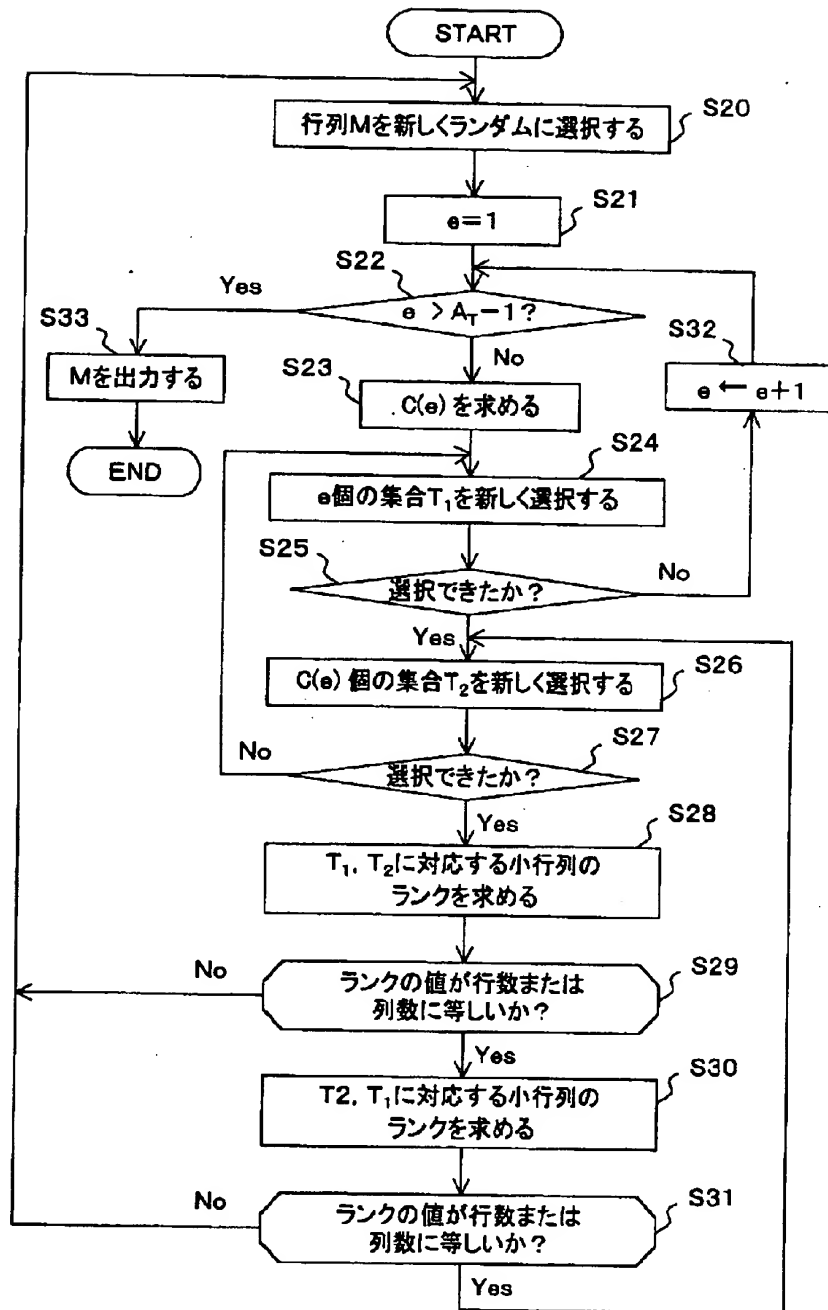
【図5】

アクティブSボックスの数の
最大値 A_T を求める処理の詳細フローチャート



【図6】

擬似MDS行列を求める処理の詳細フローチャート



【図7】

求められた擬似MDS行列の例を示す図

$$M = \begin{pmatrix} 11111111 & 11001111 & 11110111 & 10111010 & 01010100 & 00111000 \\ 11110111 & 10001111 & 11111111 & 01000101 & 10110001 & 01110000 \\ 01000111 & 00001110 & 11011111 & 10001010 & 01010101 & 11101011 \\ 00001110 & 00111000 & 10001111 & 00000101 & 11010100 & 00111111 \\ 11011000 & 00111000 & 10000100 & 10000000 & 00000000 & 11100000 \\ 00111100 & 10110111 & 01011111 & 01000000 & 11010100 & 10001000 \\ 01111100 & 01111111 & 10111010 & 10000000 & 10000101 & 00110000 \\ 01111101 & 11111100 & 01000101 & 00010101 & 00011111 & 01000000 \\ 01111111 & 11000101 & 10001010 & 01010100 & 01111100 & 00010101 \\ 01110111 & 10111111 & 00000101 & 10110000 & 11110000 & 10101010 \\ 11110000 & 11111100 & 11000101 & 01110101 & 10000101 & 10001111 \\ 01010101 & 11000101 & 10111111 & 11010100 & 00011111 & 00000110 \\ 10111111 & 10111111 & 01011111 & 10000101 & 01111100 & 00011110 \\ 11111110 & 01011111 & 10111100 & 00011111 & 11110000 & 00011000 \\ 11110001 & 10111100 & 01000101 & 01111100 & 11110101 & 01110000 \\ 10011100 & 00110000 & 01111111 & 11011111 & 11000101 & 10010101 \\ 00111000 & 01000000 & 11111100 & 10001111 & 10111111 & 10101010 \\ 11110000 & 10000000 & 11000101 & 00001111 & 01010111 & 11010100 \\ 01010101 & 00010101 & 10111111 & 00011100 & 10111100 & 00111011 \\ 00111111 & 01010100 & 01011111 & 01110000 & 01000101 & 01101010 \\ 01111101 & 11000101 & 01000101 & 11111100 & 11000000 & 11011110 \\ 11111111 & 10111111 & 10001010 & 11000101 & 10101011 & 00100011 \\ 01110111 & 10101111 & 00000101 & 10111111 & 01111111 & 01000100 \\ 11000111 & 10111100 & 00001010 & 01011111 & 11111100 & 00000010 \\ 10000111 & 01000101 & 00010000 & 10111100 & 11000101 & 00000110 \\ 11110100 & 11111111 & 11011111 & 00110000 & 01000000 & 10000100 \\ 01000011 & 11101111 & 10001111 & 01000000 & 10000000 & 10001000 \\ 10001111 & 10001111 & 00001111 & 10000000 & 00010101 & 10100000 \\ 10111100 & 00001110 & 00011100 & 00010101 & 01010100 & 01000000 \\ 11111000 & 00011100 & 01110000 & 01010100 & 10110000 & 10001011 \\ 11000011 & 11110000 & 00111100 & 11111100 & 01000101 & 11111000 \end{pmatrix}$$

【図8】

2つの集合に対応する小行列を説明する図

$$M = \begin{pmatrix} 6 & 5 & 5 & 5 & 5 & 6 \\ M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{pmatrix} \begin{matrix} 6 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 6 \end{matrix} \Rightarrow \begin{pmatrix} M_{22} & M_{23} & M_{25} & M_{26} \\ M_{32} & M_{33} & M_{35} & M_{36} \\ M_{62} & M_{63} & M_{65} & M_{66} \end{pmatrix}$$

(a)

$$M = \begin{pmatrix} 6 & 5 & 5 & 5 & 5 & 6 \\ M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{pmatrix} \begin{matrix} 6 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 6 \end{matrix} \Rightarrow \begin{pmatrix} M_{22} & M_{23} & M_{26} \\ M_{32} & M_{33} & M_{36} \\ M_{62} & M_{63} & M_{66} \end{pmatrix}$$

(b)

【図9】

擬似MDS行列の
小行列の例(その1)を示す図

$$\left(\begin{array}{cccccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array} \right)$$

【図 1 0】

擬似MDS行列の
小行列の例(その2)を示す図

1	1	1	1	1	1
1	1	1	0	1	1
0	1	0	0	1	1
0	0	0	0	1	1
0	0	0	1	1	0
1	1	0	1	0	0
<hr/>					
0	0	1	1	1	0
0	1	1	1	0	0
0	1	1	1	0	1
0	1	1	1	1	1
0	1	1	0	1	1

【図11】

30行×30列のMDS行列を求めるための部分行列を示す図

(その1)

0 : matrix[5,5]=	8 : matrix[5,5]=
00000	01000
00000	10000
00000	00101
00000	01010
00000	10100
0	1
1 : matrix[5,5]=	9 : matrix[5,5]=
00001	01001
00010	10010
00100	00001
01000	00010
10000	00100
1	1
2 : matrix[5,5]=	10 : matrix[5,5]=
00010	01010
00100	10100
01000	01101
10000	11010
00101	10001
1	1
3 : matrix[5,5]=	11 : matrix[5,5]=
00011	01011
00110	10110
01100	01001
11000	10010
10101	00001
1	1
4 : matrix[5,5]=	12 : matrix[5,5]=
00100	01100
01000	11000
10000	10101
00101	01111
01010	11110
1	1
5 : matrix[5,5]=	13 : matrix[5,5]=
00101	01101
01010	11010
10100	10001
01101	00111
11010	01110
1	1
6 : matrix[5,5]=	14 : matrix[5,5]=
00110	01110
01100	11100
11000	11101
10101	11111
01111	11011
1	1
7 : matrix[5,5]=	15 : matrix[5,5]=
00111	01111
01110	11110
11100	11001
11101	10111
11111	01011
1	1

【図12】

30行 x 30列のMDS行列を求めるための部分行列を示す図

16 : matrix[5,5]= 24 : matrix[5,5]= (その2)
 10000 11000
 00101 10101
 01010 01111
 10100 11110
 01101 11001
 1 1
 17 : matrix[5,5]= 25 : matrix[5,5]=
 10001 11001
 00111 10111
 01110 01011
 11100 10110
 11101 01001
 1 1
 18 : matrix[5,5]= 26 : matrix[5,5]=
 10010 11010
 00001 10001
 00010 00111
 00100 01110
 01000 11100
 1 1
 19 : matrix[5,5]= 27 : matrix[5,5]=
 10011 11011
 00011 10011
 00110 00011
 01100 00110
 11000 01100
 1 1
 20 : matrix[5,5]= 28 : matrix[5,5]=
 10100 11100
 01101 11101
 11010 11111
 10001 11011
 00111 10011
 1 1
 21 : matrix[5,5]= 29 : matrix[5,5]=
 10101 11101
 01111 11111
 11110 11011
 11001 10011
 10111 00011
 1 1
 22 : matrix[5,5]= 30 : matrix[5,5]=
 10110 11110
 01001 11001
 10010 10111
 00001 01011
 00010 10110
 1 1
 23 : matrix[5,5]= 31 : matrix[5,5]=
 10111 11111
 01011 11011
 10110 10011
 01001 00011
 10010 00110
 1 1

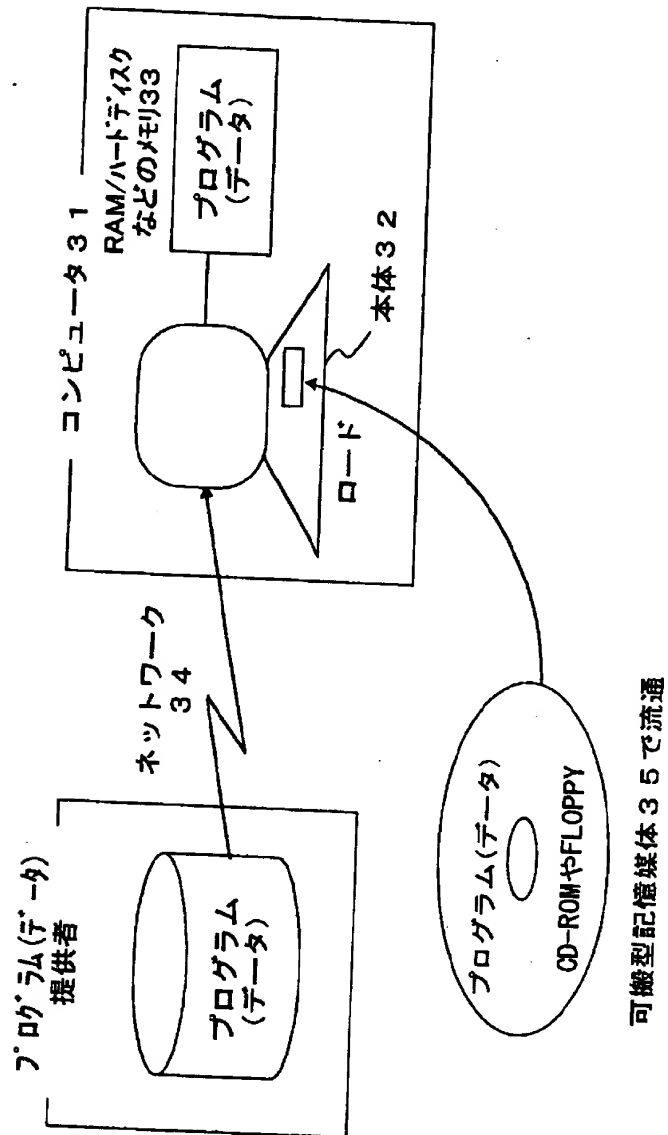
【図 1 3】

図 11, 図 12 の部分行列を用いた
MDS 行列の例を示す図

31	27	29	22	10	12
14	21	11	8	26	4
24	30	25	13	17	19
6	4	15	27	25	5
29	25	9	30	24	22
26	31	27	4	8	2

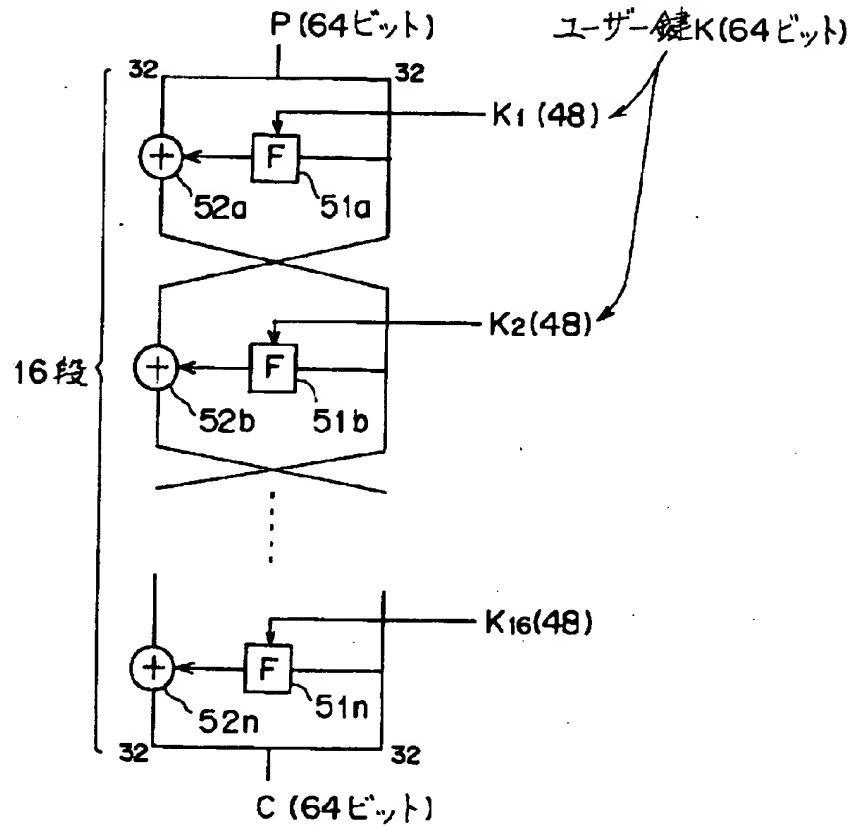
【図14】

本発明におけるプログラムの
コンピュータへのローディングを説明する図



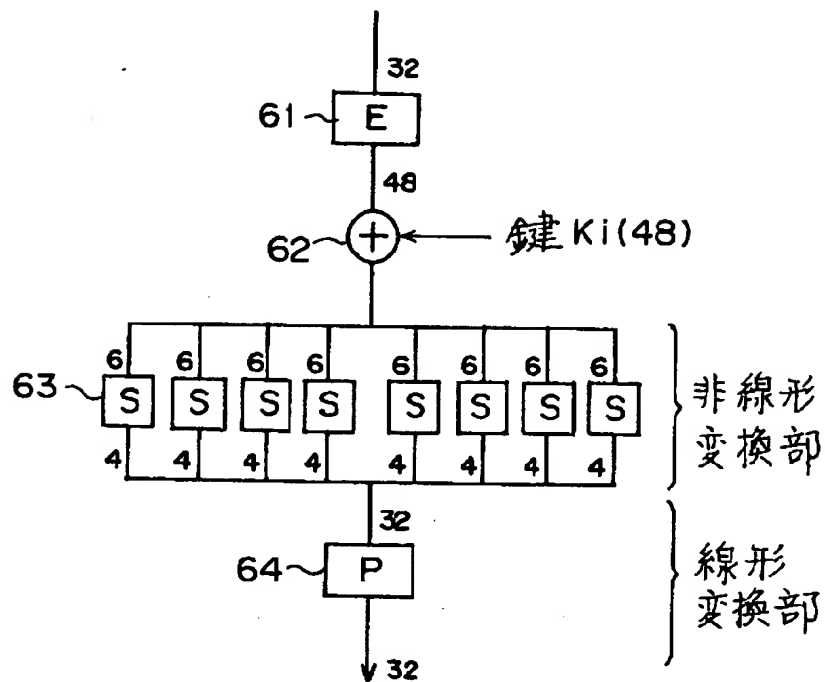
【図15】

DES 暗号の基本構造を示す図



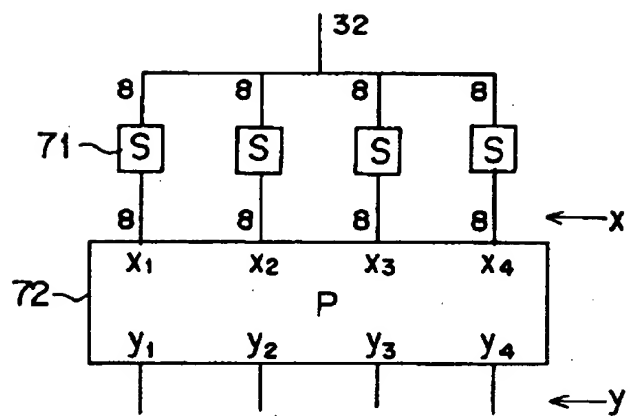
【図 16】

図 15 における F 関数の構成例の説明図



【図17】

F関数内の線形変換Pとしての
MDS変換の説明図



【図 18】

MDS変換としてのMDS行列の説明図

$$\begin{array}{c}
 \left(\begin{array}{cccc}
 a_{11} & a_{12} & a_{13} & a_{14} \\
 a_{21} & a_{22} & a_{23} & a_{24} \\
 a_{31} & a_{32} & a_{33} & a_{34} \\
 a_{41} & a_{42} & a_{43} & a_{44}
 \end{array} \right)
 \end{array}$$

32ビット

【書類名】 要約書

【要約】

【課題】 F関数内部のSPN構造中の複数のSボックスの間で入出力ビット数が同一でない場合に、SPN構造内の線形変換部としてデータ拡散性能に優れたものを用いて演算を行う。

【解決手段】 演算装置1に与えられる入力データの全ビット数を非均等に分割したビット数集合Tの入力を受け取る手段1と、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 A_T を出力する手段3を備える。さらに A_T の値が正のとき適切な線形変換部が存在すると判定する手段4と、そのような線形変換部としての擬似MDS行列を生成する手段5を備える。

【選択図】 図1

出 願 人 履 歷 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日
[変更理由] 住所変更
住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社